

IN THE CLAIMS

1. (Currently Amended) An integrated development environment-~~(999)~~ for developing user interface documents, comprising:
an editor-~~(104)~~ for editing to edit a user interface document-~~(300)~~ ;
an adaptation engine-~~(105)~~ for generating to generate device class specific representations ~~(301, 302)~~ of the user interface document-~~(300)~~, each device class specific representation ~~(301, 302)~~ referring to a respective device class-~~(DC1, DC2)~~ ;
~~characterized in that~~
the integrated development environment-~~(999)~~ further comprising a device class dependent complexity indicator-~~(121)~~ for determining to determine complexity values of layout components-~~(1 to 9)~~ of the device class specific representations ~~(301, 302)~~ by using complexity evaluation functions-~~(EF5-DC1, EF5-DC2, EF6-DC1, EF6-DC2)~~, associated with the layout components ~~(5, 6)~~ and for aggregating to aggregate the complexity values by device class according to a corresponding layout component hierarchy-~~(321, 322)~~ of the respective device class specific representation-~~(301, 302)~~.
2. (Currently Amended) The integrated development environment of claim 1, further comprising:
a template wizard-~~(106)~~ being interfaced to the editor-~~(104)~~ for creating-~~(502)~~ to create a new user interface document-~~(300)~~ by loading a predefined document template from the template wizard-~~(106)~~ into the editor-~~(104)~~.
3. (Currently Amended) The integrated development environment of claim 3, where the editor-~~(104)~~ is interfaced to a template XML description file including information about different available document templates, the information comprising meta data about device classes supported by the templates.

-
4. The integrated development environment of claim 1 ~~any one of the previous claims~~, further comprising
- a tree-based outline editor ~~(109)~~ for generating to generate an outline view ~~(209)~~ of the user interface document ~~(300)~~ when loaded into the editor ~~(104)~~, the tree-based outline editor ~~(109)~~ being interfaced to the editor ~~(104)~~ so that selection of an element ~~(209)~~ in the outline view ~~209~~ causes the editor ~~(104)~~ to highlight ~~(504)~~ a corresponding text portion ~~(309)~~ of the user interface document ~~(300)~~.
5. (Currently Amended) The integrated development environment of claim 1 ~~any one of the previous claims~~, further comprising:
- a code completion tool ~~(102)~~ for proposing to propose possibilities for auto-insertion of text in the editor ~~(104)~~ dependent on document context at a specific position within the user interface document ~~(300)~~.
6. (Currently Amended) The integrated development environment of claim 1 ~~any one of the previous claims~~, further comprising:
- a fragment repository ~~(123)~~ for saving to save from or loading to the user interface document ~~(300)~~ a document fragment having a layout that is specific to a specific device class.
7. (Currently Amended) The integrated development environment of claim 1 ~~any one of the previous claims~~, further comprising:
- a Java filtering tool ~~(108)~~ for hiding to hide Java code in the editor ~~(104)~~ when using an XML view for editing the user interface document ~~(300)~~, and for editing to edit ~~Java~~ edit Java code when activating a Java code view for editing the user interface document ~~(300)~~, wherein the editor ~~(104)~~ is configured to save the user interface document ~~(300)~~ including Java code independent from the current editing view.

-
8. (Currently Amended) The integrated development environment of claim 1 ~~any one of the previous claims~~, further comprising:
a device class dependent frames layouting view ~~(124)~~ being interfaced to the editor ~~(104)~~
~~for providing to provide~~ an overview of presentation structures of the user
interface document ~~(300)~~ for various device classes.
9. (Currently Amended) The integrated development environment of claim 1 ~~any one of the previous claims~~, further comprising:
a device class dependent page view ~~(122)~~ ~~for using to use~~ the adaptation engine ~~(105)~~ to
execute a pre-pagination run with respect to the device class specific
representations ~~(301, 302)~~ and ~~for visualizing to visualize~~ the result of the pre-
pagination run for the respective device classes ~~(DC1, DC2)~~.
10. (Currently Amended) The integrated development environment ~~(999)~~ of claim 1 ~~any one of the claims from 1 to 8~~; wherein the device class dependent complexity indicator ~~(121)~~
is replaced by
a device class dependent page view ~~(122)~~ ~~for using to use~~ the adaptation engine ~~(105)~~ to
execute a pre-pagination run with respect to the device class specific
representations ~~(301, 302)~~ and ~~for visualizing to visualize~~ the result of the pre-
pagination run for the respective device classes ~~(DC1, DC2)~~.
11. (Currently Amended) The integrated development environment ~~(999)~~ of claim 1 ~~any one of the claims from 1 to 7~~; wherein the device class dependent complexity indicator ~~(121)~~
is replaced by
a device class dependent frames layouting view ~~(124)~~ being interfaced to the editor ~~(104)~~
~~for providing to provide~~ an overview of presentation structures of the user
interface document ~~(300)~~ for various device classes.

12. (Currently Amended) The integrated development environment claim 11, further comprising:
a device class dependent page view ~~(122)~~ for using to use the adaptation engine ~~(105)~~ to execute a pre-pagination run with respect to the device class specific representations ~~(301, 302)~~ and ~~for visualizing to~~ visualize the result of the pre-pagination run for the respective device classes ~~(DC1, DC2)~~.
13. (Currently Amended) A computer implemented method for generating user interface documents, comprising ~~the steps of~~:
loading a user interface document ~~(300)~~ into an editor ~~(104)~~;
generating device class specific representations ~~(301, 302)~~ of the user interface document ~~(300)~~ by using an adaptation engine ~~(105)~~, wherein each device class specific representation ~~(301, 302)~~ refers referring to a respective device class ~~(DC1, DC2)~~;
~~characterized in that the method comprises the further steps~~ operations performed by a complexity indicator ~~(121)~~ :
determining complexity values of layout components ~~(1 to 9)~~ of the device class specific representations ~~(301, 302)~~ by using complexity evaluation functions ~~(EF5-DC1, EF5-DC2, EF6-DC1, EF6-DC2)~~, associated with the layout components ~~(5, 6)~~;
and
aggregating the complexity values by device class according to a corresponding layout component hierarchy ~~(321, 322)~~ of the respective device class specific representation ~~(301, 302)~~.
14. (Currently Amended) The method of claim 13, further comprising ~~the further step~~:
providing an overview of presentation structures of the user interface document ~~(300)~~ for various device classes.

-
15. (Currently Amended) The method of claim 13 ~~or 14~~, further comprising ~~the further steps~~:
executing a pre-pagination run with respect to the device class specific representations
(301, 302) by using the adaptation engine ~~(105)~~; and
visualizing the result of the pre-pagination run for the respective device classes (~~DC1~~,
~~DC2~~) in a device class dependent page view ~~(122)~~.
16. (Currently Amended) The method of claim 13 ~~or 14~~, wherein the determining and
aggregating ~~steps~~operations are replaced by the steps~~operations~~:
executing a pre-pagination run with respect to the device class specific representations
(301, 302) by using the adaptation engine ~~(105)~~; and
visualizing the result of the pre-pagination run for the respective device classes (~~DC1~~,
~~DC2~~) in a device class dependent page view ~~(122)~~.
17. (Canceled) ~~A computer system comprising at least one computing device having data
storage means and at least one processor to run an integrated development environment
(999) according to claim 1 any one of the claims 1 to 12.~~